01-10-00

A

jc525 U.S. PTO 09/479146 01/07/00

jc525 U.S. PTO 01/07/00

UTILITY PATENT APPLICATION TRANSMITTAL

Address to: Box PATENT APPLICATION Assistant Commissioner for Patents Washington, DC 20231	Attorney Docket No. 99-051-TAP
	Inventor(s) or Application Identifier: STEPHEN FULD

1. This application entitled METHOD AND SYSTEM FOR RECONSTRUCTING DATA IN A REDUNDANT ARRAY OF STORAGE ELEMENTS is:

   a. _X_ A new application under 37 C.F.R. §1.53(b).

   b. ___ A ___ continuation ___ divisional or ___ continuation-in-part application under 37 C.F.R. § 1.53(b) of prior application Serial No. ___/_____ filed on _____, entitled _____.

Application elements and other attached papers:

2. _X_ Specification (incl. Claims and Abstract) [Total Pages 19]

3. _X_ Drawings (___ informal _X_ formal) [Total Sheets 3]

4. _X_ Oath or Declaration

   a. _X_ Newly-executed

   b. ___ Copy from a prior application (37 C.F.R. § 1.63(d))

5. ___ Incorporation By Reference: The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Item 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.

6. ___ This application is filed by fewer than all the inventors named in the prior application, 37 C.F.R. § 1.53(d)(4).

   a. ___ ***DELETE*** the following inventor(s) named in the prior nonprovisional application:

   _____

   _____

   b. ___ The inventor(s) to be deleted are set forth on a separate sheet attached hereto.

CERTIFICATION UNDER 37 C.F.R. § 1.10

I hereby certify that this UTILITY PATENT APPLICATION TRANSMITTAL and the documents referred to as attached therein are being deposited on the below date with the United States Postal Service in an envelope as "Express Mail Post Office to Addressee" addressed to: Box Patent Application, Assistant Commissioner for Patents, Washington, D.C. 20231.

Express Mail Label No. EJ124052950US

Stephanie Klepp
(Type or print name of person mailing paper)

Date of Deposit: January 7, 2000

Stephanie Klepp
(Signature of person mailing paper)

7. Preliminary Amendment:

a. ___ A Preliminary Amendment is attached.

b. ___ Cancel in this application original claims _____ of the prior application before calculating the filing fee.

c. ___ Please amend the specification by inserting before the first line the sentence:

"This is a

___ continuation

___ divisional

of copending application(s)

___ Serial number ___ /_____ filed on _____."

d. ___ A Petition to Suspend Prosecution For The Time Necessary to File An Amendment (New Application Filed Concurrently) is attached.

8. Small entity status:

a. ___ A small entity statement is attached.

b. ___ A small entity statement was filed in the prior nonprovisional application and such status is still proper and desired.

c. ___ Is no longer desired.

9. Fee Calculation:

FOR	NUMBER FILED	NUMBER EXTRA	RATE	CALCULATIONS
TOTAL CLAIMS (37 C F R § 1 16(c))	17 -20 =		X 18.00	
INDEPENDENT CLAIMS (37 C F R § 1 16(b))	4 -3 =		X 78.00	78.00
MULTIPLE DEPENDENT CLAIMS (if applicable) (37 C.F.R §1.16(d))			260.00	
			BASIC FEE (37 C F R § 1 16(a))	690.00
		Total of above Calculations =		768.00
Reduction by 50% for filing by small entity (Note 37 C.F.R. §§ 1.9, 1.27, 1.28)				
		Assignment Recordal Fee	40.00	40.00
			TOTAL =	808.00

10. ___ A check in the amount of $_____ is enclosed.

11. _X_ The Commissioner is hereby authorized to credit overpayments or charge the following fees (or any deficiency therein) to Storage Technology Corporation Deposit Account No. 19-4545 :

a. _X_ Fees required under 37 C.F.R. § 1.16.

b. _X_ Fees required under 37 C.F.R. § 1.17.

12. Maintenance of Copendency of Prior Application

___ A request for extension of time and the appropriate fee have been filed in the pending **prior** application (or are being filed in the prior application concurrently herewith) to extend the period for response until _____.

13. _X_ An Information Disclosure Statement (IDS) is attached, along with the following indicated attachments thereto:

a. _X_ Form PTO/SB/08 (____1____ sheet(s))

b. _X_ Copies of references cited

14. ___ Certified copy of priority document(s)

15. _X_ Return Receipt Postcard

16. ___ Other: _____

_____

17. _X_ An Assignment of the invention to Storage Technology Corporation

_____

a. _X_ is attached.

b. ___ was recorded on _____ at Reel _____, Frame _____.

18. The power of attorney in the prior application is to:

_____

Name of Attorney of Record Reg. No.

___ The power appears in the original papers in the prior application.

___ The power does not appear in the original papers, but was filed on _____.

___ A new power has been executed and is attached.

19. Correspondence Address: Please address all future communications to:

Timothy R. Schulte,
Storage Technology Corporation
One StorageTek Drive, MS-4309
Louisville, Colorado 80028-4309
Telephone: 303-673-5989; Fax: 303-673-4151

Respectfully submitted,

Date 1/6/00

_____

Name: Timothy R. Schulte
Registration No.: 29,013

_X_ Attorney or agent of record
___ Filed under Rule 34(a)

99-051-TAP

METHOD AND SYSTEM FOR RECONSTRUCTING DATA IN A REDUNDANT ARRAY OF STORAGE ELEMENTS

Technical Field

The present invention relates generally to redundant array of inexpensive disks (RAID) methods and systems and, more particularly, to a data storage array method and system which accumulate a running parity of data blocks as the data blocks are read sequentially from respective storage elements such that the data blocks prior to a bad data block do not have to be reread for obtaining the parity information needed to complete the reconstruction of the bad data block.

Background Art

A data storage array is a collection of storage elements that are accessible by a host computer as a single storage unit. The individual storage elements can be any type, or a combination of types, of storage devices such as, hard disk drives, semiconductor memory, optical disks, magnetic tape drives, and the like. The individual storage elements may also be a track in a single magnetic tape.

A data storage array includes a collection of storage elements and a controller. The controller controls the operation of the storage elements and presents them as a single storage unit to a host computer. The host computer typically executes an operating system and application programs. A virtual storage element is an abstract entity realized by the controller and the data storage elements. A virtual storage element is functionally identical to a single physical storage element from the standpoint of the host computer.

One such data storage arrays is a redundant array of independent disks (RAID) or tapes (RAIT). RAID comes in various operating levels which range from RAID level 0 (RAID-0) to RAID level 6 (RAID-6). Additionally, there are multiple combinations of the various RAID levels that form hybrid RAID levels such as RAID-5+, RAID-6+, RAID-10, RAID-53 and so on. Each RAID level represents a different form of data management and data storage within the RAID disk array.

In a RAID-4 array, data is generally mapped to the various physical storage elements in data "stripes" across the storage elements and vertically in a "block" within a single storage element. To facilitate data storage, a serial data stream is partitioned into data blocks. Each data block is stored on a different storage element as the data blocks are striped across the storage elements. Once all the storage elements in a data stripe have been given data blocks, the storage process returns to the first storage element in the data stripe, and stripes data blocks across all the storage elements again.

In a RAID-4 array, data consistency and redundancy is assured using parity data that is striped to one of the storage elements. Specifically, a RAID-4 array contains N-1 data storage elements and a parity storage element. Each data stripe contains N-1 data blocks and one parity block. N-1 data blocks are striped to respective N-1 data storage elements and a parity block is striped to the parity storage element. The process then continues for the next data stripe.

RAID-4 parity is generated using an exclusive OR (XOR) function. In general, parity data is generated by taking an XOR function of the data blocks within a given data stripe. Using the parity information, the contents of any block on any single one of the storage elements in the array can be regenerated from the

contents of the corresponding blocks on the remaining storage elements in the array. Consequently, if the XOR of all corresponding blocks in the data stripe, except one is computed, the result is the remaining block. Thus, if data storage element one in the storage element array should fail, for example, the data block it contains can still be delivered to applications by reading corresponding blocks (data and parity) from all the surviving storage elements and computing the XOR of their contents. As such, the RAID-4 array is said to be fault tolerant, i.e., the loss of one storage element in the array does not impact data availability.

In a data storage array, the controller accumulates parity as the data blocks are sequentially written from the host computer to the storage elements. When all of the data blocks in a data stripe have been written to respective data storage elements, the controller writes the parity block to the parity storage element. The controller then reinitializes and starts accumulating parity again for the next data stripe to be written to the storage elements. This continues until all of the data blocks have been written to the storage elements.

During normal reading operation, the controller sequentially passes the data blocks to the host computer from the storage elements and discards the parity blocks. If a read error occurs such that a data block is unreadable, the controller returns to the storage element having the first data block in the data stripe and then sequentially rereads the data blocks prior to the bad data block. The controller accumulates parity for the reread data blocks. The controller then skips the bad data block and continues sequentially reading the remaining data blocks and the parity block in the data stripe. The controller then reconstructs the bad data block from three factors. These factors are the parity of the data blocks prior to the bad data block, the data blocks after the bad data block, and the parity block.

A problem with rereading the data blocks prior to the bad data block for reconstructing the bad data block is that this operation is time consuming. What is needed is a data storage array method and system for accumulating the parity of data blocks during sequential reading. Thus, it would not be necessary to reread the data blocks prior to a bad data block as the controller would already have the information (parity of the data blocks already read) needed to complete the reconstruction of the bad data block.

0947846-080900

Summary of the Invention

Accordingly, it is an object of the present invention to provide a data storage array method and system which accumulate parity of blocks as the blocks are sequentially read from respective storage elements such that the blocks prior to a bad data block do not have to be reread for obtaining the parity information needed to complete the reconstruction of the bad data block.

It is another object of the present invention to provide a data storage array method and system in which a controller accumulates a running parity of blocks as the blocks are sequentially read from magnetic tape to obtain the parity information needed to complete the reconstruction of a bad data block.

It is a further object of the present invention to provide a data storage array method and system which accumulate parity of data blocks as the data blocks are sequentially read from respective storage elements such that the data blocks prior to a bad data block do not have to be reread for obtaining the parity information needed to complete the reconstruction of the bad data block.

In carrying out the above objects and other objects, the present invention provides a method for reading data blocks in a data storage array having data storage elements and a parity storage element. Each of the data storage elements store a respective data block and the parity storage element stores a parity block. The parity block is based on the data blocks. The method includes reading blocks sequentially from respective storage elements. Any bad data blocks are determined as the data blocks are being read. Parity of the good data blocks is then accumulated as the data blocks are being read. The parity block is then read from the parity storage element. A bad data block is then reconstructed from the accumulated parity of the data blocks and the parity block.

In carrying out the above objects and other objects, the present invention provides another method for reading data blocks. This method includes reading data blocks sequentially from respective data storage elements. Any bad data blocks are determined as the data blocks are being read. Parity of the good data blocks is then accumulated as the data blocks are being read. The parity block is then read from the parity storage element. A bad data block is then reconstructed from the accumulated parity of the good data blocks and the parity block.

Preferably, accumulating parity of the data blocks includes exclusive ORing the parity of the good data blocks read prior to the current good data block being read with the current good data block being read. Reconstructing a bad data block includes exclusive ORing the parity of the good data blocks with the parity block. The method preferably further includes storing the good data blocks read after the bad data block until the bad data block is reconstructed to preserve ordering of the data blocks during reading.

In carrying out the above objects and other objects, the present invention further provides a data storage array for reading data blocks. The system includes data storage elements each storing a respective data block and a parity storage element storing a parity block based on the data blocks. A controller sequentially reads data blocks and the parity block from respective storage elements. The controller determines if any of the data blocks are bad as the data blocks are being read. A parity accumulator accumulates parity of the good data blocks as the controller reads the data blocks until the controller determines one of the data blocks to be bad. The controller reconstructs a bad data block from the accumulated parity of the good data blocks and the parity block. The storage elements are preferably either magnetic tape drives, magnetic disk drives, or a single track of a magnetic tape.

The advantages accruing to the present invention are numerous. For instance, the present invention allows higher speed operation in the event of read errors. Additional buffer bandwidth is not required as determining the accumulated or partial parity of the blocks is the same information that is performed during writing.

The above objects and other objects, features, and advantages of the present invention are readily apparent from the following detailed description of the best mode for carrying out the present invention when taken in connection with the accompanying drawings.

Brief Description of the Drawings

FIG. 1 illustrates a block diagram of a conventional data storage array system;

FIG. 2 illustrates a conventional data striping format for a RAID-4 storage element array;

FIG. 3 illustrates a data storage array system in accordance with the present invention;

FIG. 4 illustrates a flow chart showing operation of the method and system of the present invention; and

FIG. 5 illustrates a data storage array system in accordance with an alternate embodiment of the present invention.

Best Modes For Carrying Out The Invention

Referring now to FIG. 1, a data storage array system 10 containing a host computer 12 and a plurality of individual storage elements 14a, 14b, 14c, and 14d is shown. Host computer 12 executes, on one or more processors, an operating system as well as various applications that periodically read and write information from and to the storage elements. For simplicity, the combination of the operating system and applications shall be referred to as the operating environment. The operating environment accesses various ones of the storage

elements as if the individual storage elements functioned as a single, large storage element.

Typically, the storage elements are inexpensive hard disk drives arranged in an array. Various other forms of data storage elements may be substituted for, or combined with, the disks. Such other forms of storage elements include, but are not limited to, optical drives, ZIP drives, floppy disk drives, semiconductor memory devices, magnetic tape drives, or any other form of random access memory. Although any of these other forms of storage elements may be used with the present invention, the following discussion refers to a single track of a single magnetic tape, i.e., single tape RAIT.

When using a single magnetic tape, the operating environment treats the single track 14 of the magnetic tape (shown in FIG. 3) commonly referred to as a virtual tape 16. The data blocks are represented to host computer 12 as being stored in virtual tape 16 in serial fashion. Thus, the operating environment requests data from virtual tape 16 using a convenient single address. A controller 18 ensures that the address into virtual tape 16 is mapped to an address within the single track of the magnetic tape. Controller 18 can also reside within host computer 12.

In a redundant array of independent storage devices, such as disks or tapes, the physical storage units 14a, 14b, 14c, and 14d, known as members, are accessed via controller 18. Data blocks are distributed, or striped, serially amongst members 14a, 14b, 14c, and 14d. To achieve fault tolerance, the data blocks are arranged into redundancy groups where the data blocks within a data stripe are self consistent with the other data blocks in the data stripe. One technique for making the data stripe redundant is to use a parity block based on the

data blocks. As such, the redundancy information is a parity block in which the data blocks in the data stripe are XOR (exclusive OR) consistent.

FIG. 2 depicts a conventional data striping format for a RAID-4 storage element array. Under the RAID-4 standard, data blocks appearing in virtual track 16 are striped across the members 1-3. A parity block based on the data blocks is striped to member 4. Under RAID-4, the parity block is stored in a dedicated parity storage member. However, the parity block may be stored in a diagonal manner across the members as done under RAID-5. The pattern of storage is not important as long as the parity block locations are mappable. Consequently, controller 18 knows where the parity blocks are stored such that the parity blocks can be easily identified.

Use of parity blocks to form consistent data stripes is well-known in the art and a description regarding the use of parity information in a RAIT-4 storage array system will clarify operation of the present invention. Parity blocks are generated by using an exclusive OR (XOR) function upon a series of data blocks located within a given data stripe 20. For example, in FIG. 2, parity block #1 is formed by applying the XOR function to data block #1 through data block #3 (e.g., data block #1 $\oplus$ data block #2 $\oplus$ data block #3 = parity block #1). Similarly, in data stripe 22, data block #4 $\oplus$ data block #5 $\oplus$ data block #6 = parity block #2 and so on. Thus, each data stripe is XOR consistent.

If one of the members ceased to function, the information contained in the remaining members is used to reconstruct the data block on the damaged member. For example, if member 2 were damaged, data block #2 in data stripe 20 would be unreadable. However, data block #2 = data block #1 $\oplus$ data block

#3 $\oplus$ parity block #1. Thus, the array is fault tolerant, i.e., if one member is damaged at a given time the system is still operational without any loss of data.

Referring now to FIGS. 3-4, with continual reference to FIGS. 1-2, the data storage array method and system of the present invention will be described. Data storage array system 30 includes a host computer 12, a single magnetic tape storage element 14, and a controller 18. Controller 18 is operable to serially write data blocks and parity blocks from host computer 12 onto a track of magnetic tape 14. Controller 18 is also operable to sequentially read data blocks and parity blocks from a track of magnetic tape 14 for host computer 12.

Controller 18 includes a buffer 32 and a parity accumulator 34. Parity accumulator 34 accumulates parity as controller 18 sequentially writes the data blocks from host computer 12 to magnetic tape 14 via buffer 32. After controller 18 sequentially writes all of the data blocks in a data stripe to magnetic tape 14, the controller writes the parity block from parity accumulator 34 to the magnetic tape. Parity accumulator then reinitializes and starts accumulating parity again for the next data stripe to be written to magnetic tape 14. This continues until controller 18 has written all of the data blocks to the single track of magnetic tape 14.

During reading, controller 18 serially passes the data blocks to host computer 12 from magnetic tape 14 via buffer 32. Controller 18 discards the parity block belonging to a data stripe after sequentially passing all of the data blocks in the data stripe to host computer 12. Controller 18 sequentially reads the data blocks and determines if any one of the data blocks are bad before passing the data blocks to host computer 12. If the data block is good, controller 18 passes the good data block to host computer 12. If the data block is bad, controller 18

reconstructs the bad data block to make it a good data block before passing it and any other good data blocks to host computer 12.

The procedure for reconstructing a bad data block into a good data block in accordance with the present invention will now be explained in greater detail. While controller 18 reads a data block, parity accumulator 34 accumulates the parity of the data blocks up to and including the current data block being read. For instance, if controller 18 has read data block #1 and is now reading data block #2, parity accumulator 34 accumulates the parity of data blocks #1 and #2 (e.g., accumulated parity = data block #1 $\oplus$ data block #2). Similarly, if controller 18 is reading data block #4, the first data block in data stipe 22, and has not read any other data blocks in data stripe 22, then parity accumulator 34 accumulates the parity of data block #4 (e.g., accumulated parity = data block #4). The accumulated parity is the partial parity of all of the blocks in a data stripe, i.e., the parity of the blocks in a data stripe that have been read.

When controller 18 determines that a data block is unreadable or bad, the controller skips the bad data block and continues sequentially reading the remaining data blocks and the parity block in the data stripe. Controller 18 then reconstructs the bad data block from two factors. The first factor is the accumulated parity of the good data blocks of the data stripe read by controller 18. Parity accumulator 34 accumulates and maintains the accumulated parity of the good data blocks as the data blocks are being read. The second factor is the parity block of the data stripe.

For instance, if a data stripe being read has six data blocks and one parity block and if data block #3 is bad, then controller 18 reconstructs data block #3 in accordance with the following equation: data block #3 = accumulated parity

$\oplus$ parity block #1, where accumulated parity = data block #1 $\oplus$ data block #2 $\oplus$ data block #4 $\oplus$ data block #5 $\oplus$ data block #6.

Buffer 32 stores the good data blocks read by controller 18 after the bad data block. After controller 18 reconstructs the bad data block and passes it to host computer 12, the controller directs buffer 32 to pass the stored good data blocks to host computer 12 in the order that they were read by the controller. This preserves data block ordering.

Referring now to FIG. 4, with continual reference to FIGS. 1-3, a flow chart 40 illustrating operation of the data storage array method and system of the present invention are shown. Flow chart 40 initially begins with controller 18 reading the data blocks sequentially from magnetic tape 14 as shown in box 42. Controller 18 then determines if any of the data blocks are bad as the data blocks are being read as shown in box 44. Parity accumulator 34 accumulates parity of the good data blocks as the data blocks are being read as shown in block 46. Controller 18 then sends the good data blocks prior to a bad data block to host computer 12 as shown in box 47.

Controller 18 then reads the parity block from magnetic tape 14 as shown in box 48. Controller 18 then reconstructs a bad data block from the accumulated parity of the good data blocks and the parity block as shown in box 50. Controller 18 then sends the reconstructed data block to host computer 12 as shown in box 52 and then sends the good data blocks after the reconstructed data block to host computer 12 as shown in box 54.

Referring now to FIG. 5 with reference to FIGS. 1-3, an alternative embodiment of the data storage array system in accordance with the present

invention is shown. Data storage array system 60 differs from data storage array system 30 shown in FIG. 3 in that separate and distinct storage elements 62a, 62b, 62c, and 62d have been substituted for magnetic tape 14. Storage elements 62 may be disk drives, tape drives, and the like. Storage elements 62 are connected in parallel to controller 18 for concurrent access with the controller. Controller 18 writes to and reads from storage elements in the same manner as described with reference to the track of magnetic tape 14 in FIG. 3.

Thus it is apparent that there has been provided, in accordance with the present invention, a data storage array method and system that fully satisfy the objects, aims, and advantages set forth above. While the present invention has been described in conjunction with specific embodiments thereof, it is evident that many alternatives, modifications, and variations will be apparent to those skilled in the art in light of the foregoing description. Accordingly, it is intended to embrace all such alternatives, modifications, and variations as fall within the spirit and broad scope of the appended claims.

What Is Claimed Is:

1. In a data storage array having data storage elements and a parity storage element, wherein each of the data storage elements store a respective data block and the parity storage element stores a parity block, the parity block being based on the data blocks, a method for reading data blocks, the method comprising:

reading data blocks sequentially from respective data storage elements;

determining if any of the data blocks are bad as the data blocks are being read;

accumulating parity of the good data blocks as the data blocks are being read;

reading the parity block from the parity storage element; and

reconstructing a bad data block from the accumulated parity of the good data blocks and the parity block.

2. The method of claim 1 wherein:

accumulating parity of the good data blocks includes exclusive ORing the parity of the good data blocks read prior to the current good data block being read with the current good data block being read.

3. The method of claim 2 wherein:

reconstructing a bad data block includes exclusive ORing the accumulated parity of the good data blocks with the parity block.

4. The method of claim 1 further comprising:

storing the good data blocks read after the bad data block until the bad data block is reconstructed to preserve ordering of the data blocks during reading.

5. A data storage array system for reading data blocks, the system comprising:

data storage elements each storing a respective data block;

a parity storage element storing a parity block based on the data blocks;

a controller for reading data blocks sequentially from respective data storage elements and the parity block, the controller operable for determining if any of the data blocks are bad as the data blocks are being read; and

a parity accumulator for accumulating parity of the good data blocks as the controller reads the data blocks, wherein the controller reconstructs a bad data block from the accumulated parity of the good data blocks and the parity block.

6. The system of claim 5 wherein:

the parity accumulator accumulates parity of the good data blocks as the data blocks are being read by exclusive ORing the accumulated parity of the good data blocks read prior to the current good data block being read with the current good data block being read.

7. The system of claim 6 wherein:

the controller reconstructs a bad data block by exclusive ORing the accumulated parity of the good data blocks with the parity block.

8. The system of claim 5 further comprising:

a buffer for storing the good data blocks read by the controller after the bad data block until the controller reconstructs the bad data block to preserve ordering of the data blocks during reading.

9. The system of claim 5 wherein:
the storage elements are magnetic tape drives.

10. The system of claim 5 wherein:
the storage elements are magnetic disk drives.

11. The system of claim 5 wherein:
the storage elements are a track of a magnetic tape.

12. In a magnetic tape having data blocks and a parity block serially arranged, the parity block being based on the data blocks, a method for reading data blocks, the method comprising:
reading data blocks sequentially from the magnetic tape;
determining if any of the data blocks are bad as the data blocks are being read;
accumulating parity of the good data blocks as the data blocks are being read;
reading the parity block from the magnetic tape; and
reconstructing a bad data block from the accumulated parity of the data blocks and the parity block.

13. The method of claim 12 wherein:

accumulating parity of the good data blocks includes exclusive ORing the parity of the good data blocks read prior to the current good data block being read with the current good data block being read.

14. The method of claim 13 wherein:

reconstructing a bad data block includes exclusive ORing the parity of the good data blocks with the parity block.

15. The method of claim 12 further comprising:

storing the good data blocks read after the bad data block until the bad data block is reconstructed to preserve ordering of the data blocks during reading.

16. A data storage array system for reading data blocks, the system comprising:

magnetic tape having data blocks and a parity block based on the data blocks;

a controller for reading data blocks sequentially from the magnetic tape and for reading the parity block from the magnetic tape, the controller operable for determining if any of the data blocks are bad as the data blocks are being read; and

a parity accumulator for accumulating parity of the good data blocks as the controller reads the data blocks, wherein the controller reconstructs a bad data block from the accumulated parity of the good data blocks and the parity block.

17. The system of claim 16 further comprising:

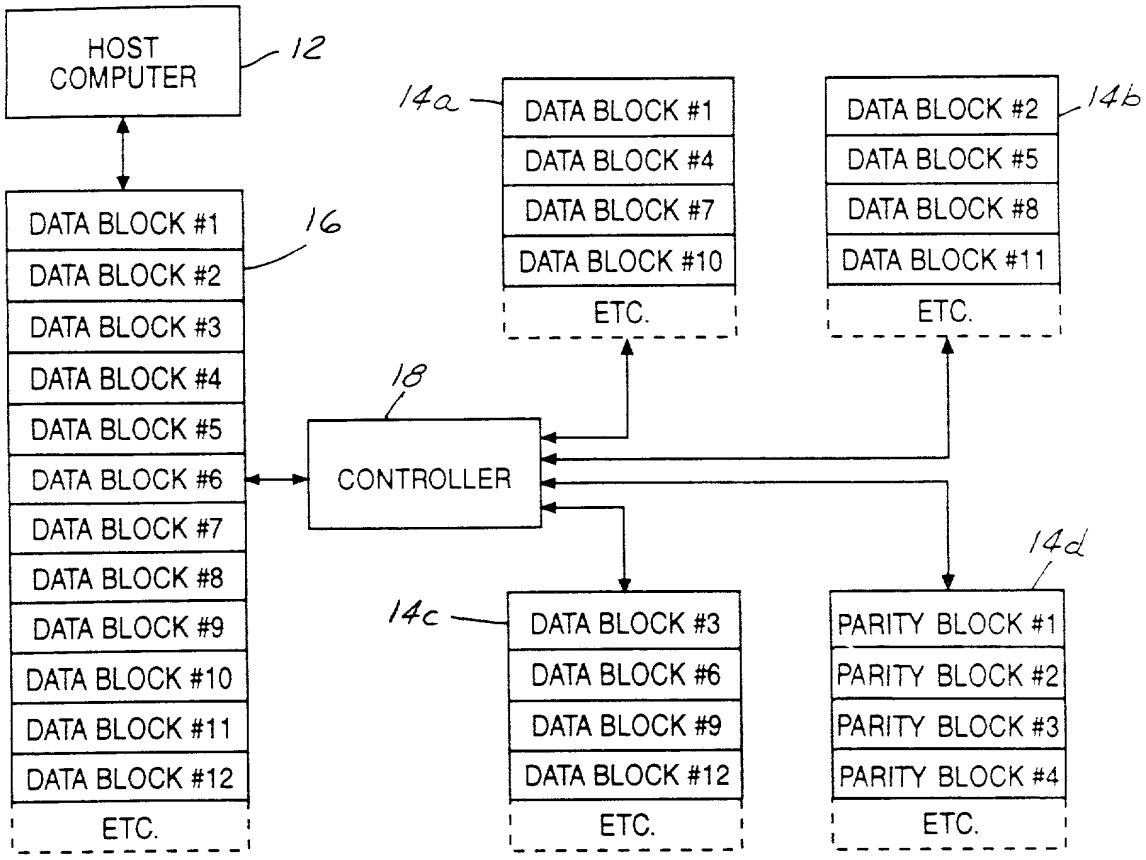
a buffer for storing the good data blocks read by the controller after the bad data block until the controller reconstructs the bad data block to preserve ordering of the data blocks during reading.

Docket #99-051-TAP

Abstract

A RAID or RAIT type data storage array such as a single tape RAIT includes storage elements storing data blocks and a parity block based on the data blocks. The array reads the data blocks sequentially from respective storage devices. The array determines if any of the data blocks are bad as the data blocks are being read. The array accumulates the parity of good data blocks as the data blocks are being read. A bad data block is reconstructed from the accumulated parity of the good data blocks and the parity block.

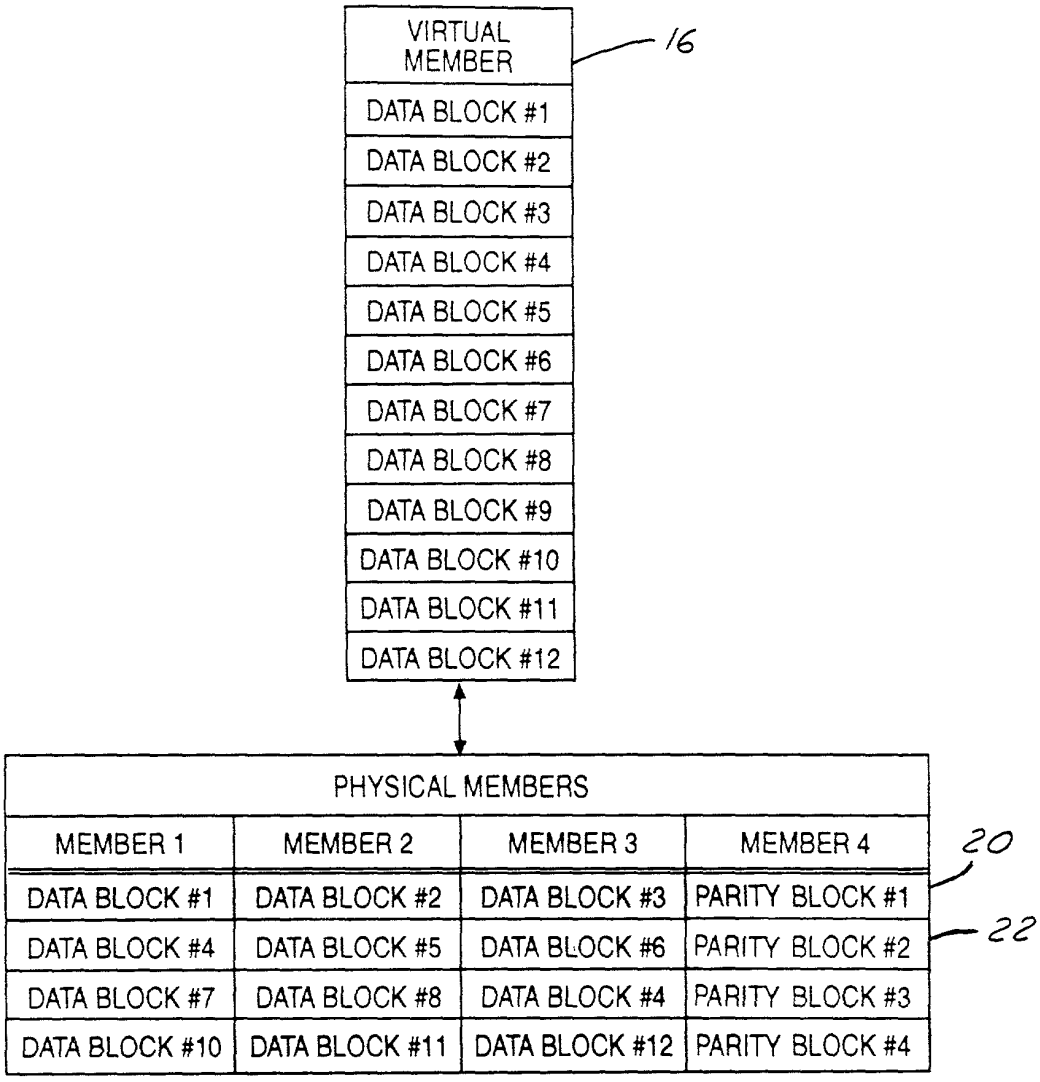
D6E7B4F5-B4C9-4D0D



(PRIOR ART)

FIG. 1

05/30/96,01:06:06



(PRIOR ART)

FIG.2

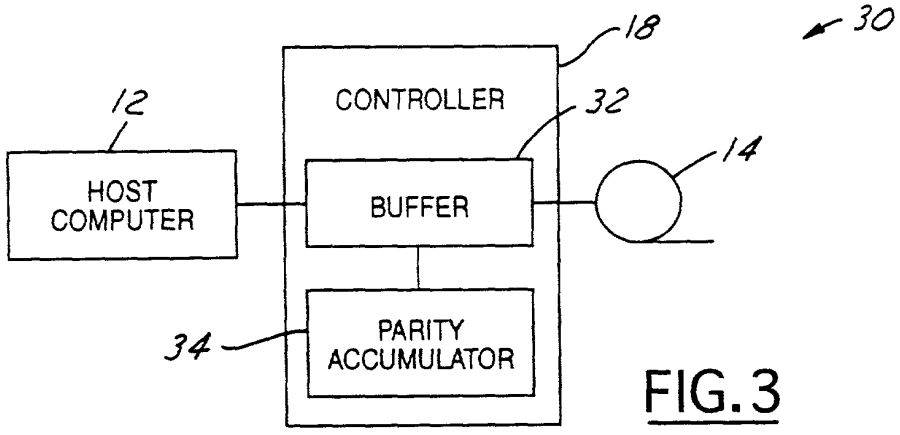


FIG.3

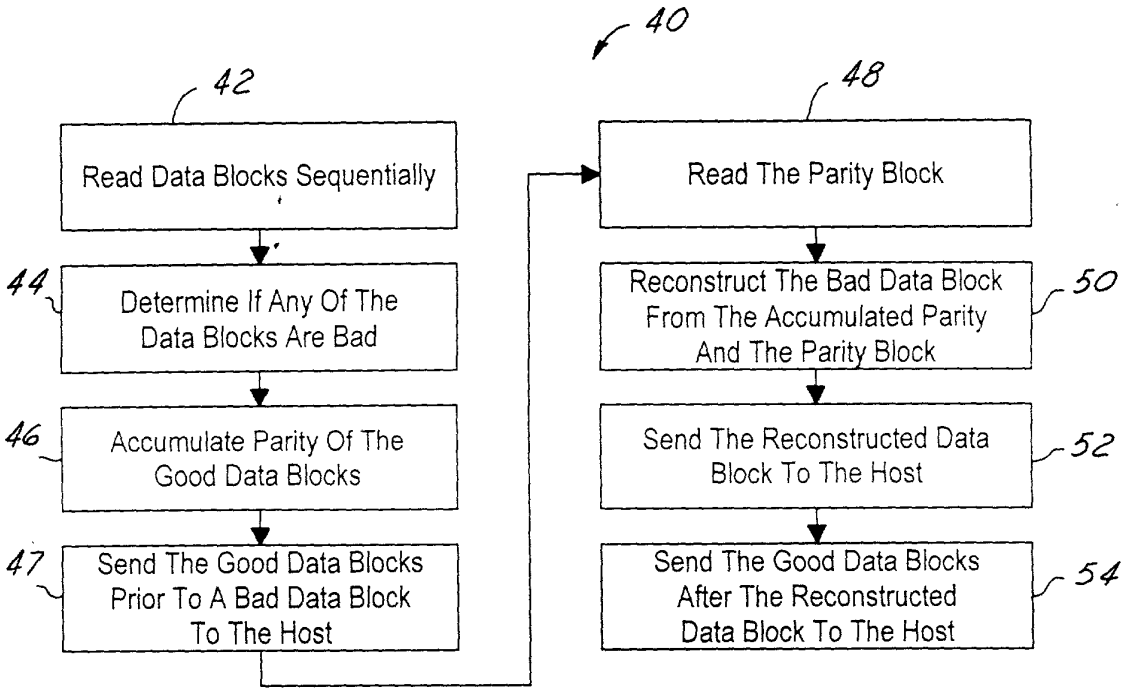

40
42
Read Data Blocks Sequentially
44
Determine If Any Of The Data Blocks Are Bad
46
Accumulate Parity Of The Good Data Blocks
47
Send The Good Data Blocks Prior To A Bad Data Block To The Host
48
Read The Parity Block
Reconstruct The Bad Data Block From The Accumulated Parity And The Parity Block
50
Send The Reconstructed Data Block To The Host
52
Send The Good Data Blocks After The Reconstructed Data Block To The Host
54

FIG. 4

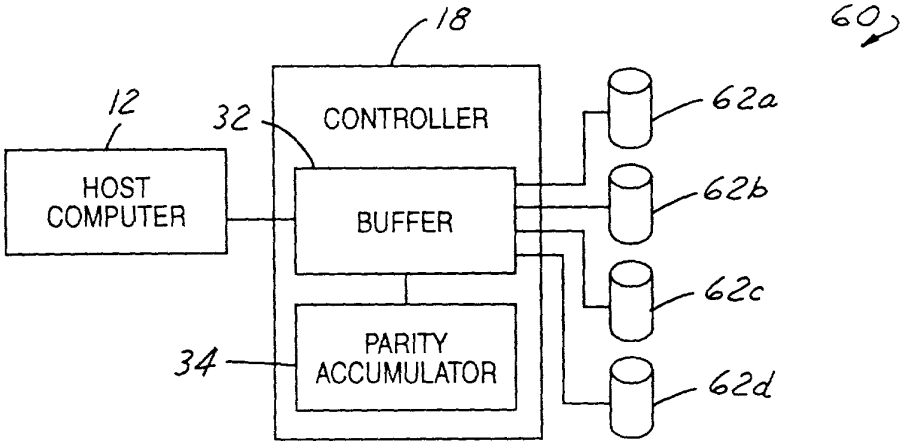

18
60
12
32
CONTROLLER
HOST COMPUTER
BUFFER
62a
62b
62c
62d
34
PARITY ACCUMULATOR

FIG. 5

DECLARATION FOR PATENT APPLICATION AND POWER OF ATTORNEY

Atty. Docket No. 99-051-TAP
First Named Inventor STEPHEN FULD

As a below named inventor, I hereby declare that my residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

METHOD AND SYSTEM FOR RECONSTRUCTING DATA IN A REDUNDANT ARRAY OF STORAGE ELEMENTS,

the specification of which:

[X] is attached hereto; or
[ ] was filed on (MM/DD/YYYY) _____ as U.S. Application Number or PCT International Application Number _____, and was amended on (MM/DD/YYYY) _____ (if applicable).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment specifically referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby appoint the following registered practitioners to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

Ernie L. Brooks, Reg. No. 26,260; James A Kushman, Reg. No. 25,634, David R Syrowik, Reg. No 27,956, Mark A. Cantor, Reg. No 30,614; Ralph M. Burton, Reg. No. 17,748; Robert C.J. Tuttle, Reg No 27,962; Earl J LaFontaine, Reg No. 30,766; Ronald M. Nabozny, Reg No. 28,648; Thomas A Lewry, Reg No. 30,770; John E Nemazi, Reg. No 30,876, Kevin J. Heinl, Reg. No. 29,805; William G Abbatt, Reg No 31,936, Donald J. Harrington, Reg. No 17,427, Paul M Schwartz, Reg No 33,278, Timothy G. Newman, Reg No 34,228, Frederick M. Ritchie, Reg. No. 18,669, Robert C Brandenburg, Reg No 29,048, A Frank Duke, Reg. No. 20,937, John M Halan, Reg. No. 35,534, Jeffrey M Szuma, Reg No 35,700; James R. Ignatowski, Reg. No. 26,741, Frank A Angileri, Reg No 36,733; William G. Conger, Reg No. 31,209; Sangeeta G. Shah, Reg No 38,614, Christopher W Quinn, Reg. No. 38,274; Robert C. Jones, Reg. No. 35,209; David S Bir, Reg No 38,383, Konstantine J. Diamond, Reg No. 39,657; James N Kallis, Reg. No. 41,102, Hugo A Delevie. Reg No 32,688, Ralph E. Smith, Reg. No. 35,474, Michael S Brodbine, Reg. No. 38,392, Jeremy J Curcuri, Reg. No. 42,454, Mark D. Chuey, Reg. No. 42,415, and John J Ignatowski, Reg. No 36,555, Pete N Kiousis, Reg. No. 41,117; Gigette M Bejin, Reg No 44,027, Stephanie M. Mansfield, Reg No 43,773, Mark E Stuenkel, Reg No 44,364, Matthew R. Mowers, Reg No P-44-956, Timothy R Schulte, Reg No. 29.013; Wayne P. Bailey, Reg No 34,289

Address all correspondence and telephone calls to Timothy R. Schulte at Storage Technology Corporation, One StorageTek Drive, MS-4309, Louisville, Colorado 80028-4309, (303) 673-5989.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Sole or First Inventor STEPHEN FULD

Inventor's signature Stephen Fuld Date Jan 5, 2000

Post Office Address 615 Valley Oak Lake, Thousand Oaks, California 91320

Residence (same as above) Citizenship USA